

VB开发CAI课件窗口的常见处理方法

方 刚

(广东技术师范学院计算机科学系 广东 广州510665)

摘要：本文介绍了在利用VB进行CAI设计中，对CAI课件窗口的处理的几种方法，并给出相关的实例。

关键字：CAI 课件 窗口 控件 API

中图分类号：G434

随着计算机的迅猛发展，计算机辅助教学CAI (Computer Assisted Instruction) 走入寻常课堂已成历史的必然，CAI是指用计算机帮助或代替教师执行部分教学任务，向学生传授知识和提供技能训练，直接为学生服务。而对于用在辅助我们教学任务的计算机程序称为教学软件或课程软件，简称为课件。一个成功的课件，离不开成功的窗口设计。窗口是应用程序界面的核心，是用户与应用程序交互的场所。Visual Basic (简称VB) 作为一种可视化编程软件，采用了面向对象的程序设计思想，以其优秀的图形化界面设计风格而著称，它继承了Basic简单易学的特点，且功能强大等，它和许多课件开发平台相比，不仅有更好的灵活性，而且还可以充分的实现开发者的意图，所以VB被许多课件的开发者所青睐，下面是本人在VB开发课件过程中，对窗口常用的处理方法。

1. 窗口在屏幕中央

我们在CAI课件中，很多时候都需要将相关的窗口显示在计算机屏幕的中央。实际上我们可以利用下面的自定义函数CenterWin，在任何需要窗口居中的地方只需加入代码“centerwin me”即可以使窗口定位在屏幕中央，具体的centerwin函数如下：

```
Public Sub CenterWin (f As Form)
f.Top =(Screen.Height*.5) - (f.Height*.5)
f.Left =(Screen.Width*.5) - (f.Width*.5)
End Sub
```

2. 窗口以全屏显示

有时CAI课件需要窗口自动以当前计算机屏幕的大小尺寸、分辨进行全屏显示时，可以采用如下方法：

```
Private Sub Form_Load()
Form1.top=0 ‘设置窗口上边界位置
Form1.left=0 ‘设置窗口左边界位置
Form1.Width=Screen.Width ‘设置窗口宽度为屏幕宽度
Form1.Height=Screen.Height ‘设置窗口高度为屏幕高度
End Sub
```

3. 窗口保持在最前面

有时CAI课件需要一个信息或查询窗口保持在最前面，可以通过调用API函数 SetWindowPos来实现，示例如下：

```
Option Explicit
Private Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As
```

```

Long
Private Const HWND_TOPMOST& =-1
' 将窗口置于列表顶部, 并位于任何最顶部窗口的前面
Private Const SWP_NOSIZE& =&H1
' 保持窗口大小
Private Const SWP_NOMOVE& =&H2
' 保持窗口位置
Private Sub Form_Load()
SetWindowPos Me.hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE Or SWP_NOSIZE
' 将窗口设为总在最前
End Sub

```

4 移动没有标题栏的窗口

在实际应用中, CAI课件有时需要没有标题栏的窗口。对于有标题栏的窗口, 我们一般是用鼠标按住窗口的标题栏, 然后移动窗口, 当窗口没有标题栏时, 相应的标题栏所具有比如移动、最大化或最小化等功能已经完全丧失。其实, 借助WINDOWS 提供的API 函数ReleaseCapture和SendMessage, 这个问题便可迎刃而解。SendMessage函数实际上是一个WINDOWS 消息处理函数, 当用于向窗口发出控制命令时, 其第一个参数为窗口句柄hwnd, 第二个参数为WM_SYSCOMMAND(&H112), 第三个参数则为发出命令的具体内容, 第四个参数恒为0&, 常用的系统命令为:

我们可以用下面的方法来移动窗口:

在 BAS 文件中声明:

```

Declare Function ReleaseCapture Lib "user32" () As Long
Declare Function SendMessage Lib "user32" _ Alias "SendMessageA" ( _ ByVal hwnd As Long, ByVal wParam As Long, _ ByVal wParam As Long, lParam As Any) As Long
Public Const HTCAPTION = 2
Public Const WM_NCLBUTTONDOWN = &HA1

```

然后, 在 Form_MouseDown 事件中:

```

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
ReleaseCapture
SendMessage hwnd, WM_NCLBUTTONDOWN, HTCAPTION, 0&
End Sub

```

程序运行后, 只要当光标落在Form区域时按住鼠标左键, 便可以拖动窗口了。

5制作“闪烁标题栏”窗口

在一些CAI课件中, 利用WINDOWS 提供的API 函数FlashWindow, 实现闪烁标题栏可以引起用户的注意, 可以取得很好的效果。

具体步骤如下:

一、创建一个新工程, 将如下代码加入到Form1的“通用\声明”中:

```

Option Explicit
Private Declare Function FlashWindow Lib "user32"(ByVal hwnd As Long, ByVal bInvert As Long) As Long

```

二、向窗口Form1中加入一个定时器控件Timer1, 设置如下属性:

```

Enabled: True
Interval: 500

```

将下列代码加入到Timer1的Timer事件中:

```

Private Sub Timer1_Timer()
FlashWindow hwnd, True

```

6. 创建特殊形状窗口

窗口的形状通常是矩形的，而在CAI课件中，适当改变窗口的形状，不但可以增加程序的生动、趣味性，而且还可以给人赏心悦目的感觉。能够完成此任务的函数为SetWindowRgn，其共有三个参数，第一个指定被剪切的对象的句柄，比如Picture图形框等，如果指定为Form则即对应用程序窗口本身进行处理，第二个参数指明剪切的形状，即指定的几何图形特征，此参数也必须由相应的API函数提供说明，第三个参数是一布尔变量，一般可设置为真(True)；在API中有多个几何图形的说明函数，常见的有：

CreateRectRgn：建立矩形区域，其参数分别为矩形的左上角坐标及右下角坐标；

CreateRoundRectRgn：建立圆角矩形区域，其参数分别为左上角及右下角坐标，还有圆角直径等，当圆角直径接近或超过矩形的长度时，将呈现为圆或椭圆形；

CreateEllipticRgn：建立椭圆矩形区域，参数分别为横向与纵向直径的起点和终点；

CreatePolygonRgn：建立多边形区域，参数比较复杂；

要完成一个特殊形状窗口的设置，需要使用区域设置函数setWindowRgn和一个几何图形说明函数，比如一个完整的圆形窗口建立命令应该为：

```
SetWindowRgn form1.hWnd, CreateEllipticRgn(10, 0, 200, 200), True
```

利用此方法得到特殊形状的窗口。又如我们分别利用CreateRoundRectRgn函数用于创建一个圆角矩形区域；CreateEllipticRgn用于创建一个椭圆区域；CombineRgn函数用于将两个区域组合为一个新区域；SetWindowRgn函数窗口的区域。

```
Private Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long,
ByVal X2 As Long, ByVal Y2 As Long) As Long
```

```
Private Declare Function CreateRoundRectRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long,
ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long) As Long
```

```
Private Declare Function CombineRgn Lib "gdi32" (ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long,
ByVal hSrcRgn2 As Long, ByVal nCombineMode As Long) As Long
```

```
Private Declare Function SetWindowRgn Lib "user32" (ByVal hWnd As Long, ByVal hRgn As Long,
ByVal bRedraw As Boolean) As Long
```

```
Private Sub Form_Load()
```

```
Dim p, q As Long
```

```
p = CreateEllipticRgn(0, 0, 150, 200)
```

```
'通过制定左上角和右下角的坐标建立一个圆
```

```
q = CreateRoundRectRgn(40, 120, 340, 420, 20, 20)
```

```
'通过制定左上角和右下角的坐标以及圆角的宽度建立一个圆角矩形
```

```
CombineRgn p, q, p, RGN_OR '联合两个区域,使用或(就是取并集)运算
```

```
SetWindowRgn hWnd, p, True '设置窗口区域
```

```
Me.Refresh '刷新窗口
```

```
End Sub
```

7. 同步缩放控件和窗口

在课件开发过程中，有时一个控件与窗口之间在设计时原本比例可以很协调，但当窗口的尺寸缩放时，窗口内的控件的尺寸以及在窗口内的相对位置如果没有进行相应的调整，那么控件与窗口就会变得很不协调。其实，我们可以通过调整窗口内控件的尺寸和调整窗口内的控件在窗口中的相对位置来保持原来的协调关系。

(1) 当窗口尺寸调整后，窗口内控件的尺寸可以按以下公式进行调整：

调整后的控件的Width值=(控件原Width值/窗口原Width值)×窗口.ScaleWidth；

调整后的控件的Height属性值=(控件原Height值/窗口原Height值)×窗口.ScaleHeight。

(2) 窗口尺寸缩放的同时，窗口内的控件在窗口中的相对位置可以按以下公式进行调整：
调整后控件的Left值=(控件原Left值/窗口原Left值)×窗口.ScaleWidth；
调整后控件的Top值=(控件原Top值/窗口原Top值)×窗口.ScaleHeight。

假设窗口上有4个控件，下面代码实现了控件和窗口同步缩放。

```
Dim frmWidth As Integer '窗体在设计状态下的宽度比例
Dim frmHeight As Integer '窗体在设计状态下的高度比例
Dim ctrWidth(1 To 4) As Integer '设计状态下窗体上所有控件的宽度
Dim ctrHeight(1 To 4) As Integer '设计状态下窗体上所有控件的高度
Dim ctrTop(1 To 4) As Integer '设计状态下所有控件最顶端在窗体上的纵坐标
Dim ctrLeft(1 To 4) As Integer '设计状态下所有控件最顶端在窗体上的纵坐标
Private Sub Form_Load()
Dim I As Integer
Dim mControl As Control
frmWidth=Me.ScaleWidth
frmHeight=Me.ScaleHeight
I=1
For Each mControl In Me.Controls
ctrWidth(I)=mControl.Width
ctrHeight(I)=mControl.Height
ctrTop(I)=mControl.Top
ctrLeft(I)=mControl.Left
I=I+1
Next
End Sub

Private Sub Form_Resize()
Dim I As Integer
Dim hScale As Double
Dim vScale As Double
Dim mControl As Control
'利用窗体的新尺寸和原始尺寸相除得到窗体的横向变化比例和纵向变化比例
hScale=ScaleWidth/frmWidth
vScale=ScaleHeight/frmHeight
I=1
For Each mControl In Me.Controls
mControl.Width=ctrWidth(I)*hScale
mControl.Height=ctrHeight(I)*vScale
mControl.Top=ctrTop(I)*vScale
mControl.Left=ctrLeft(I)*hScale
I =I+1
Next
End Sub
```

8. 添加窗口背景音乐

Windows系统中支持两种声音文件：WAV格式文件和MIDI格式文件，一般使用API(Windows应用程序编程接口)函数sndPlaySound播放WAV格式文件，使用mciExecute函数来播放MIDI文件。

(1) 使用函数sndPlaySound播放WAV格式文件

sndPlaySound函数需要两个参数：第一个参数lpstrSoundName是要播放的WAV文件的名称；第二个参数是一个表明播放方式的标识常量。

首先，必须在程序窗口的通用声明部分或标准模块中添加如下的声明：

```
Public Const SND_SYNC=&&H0 ‘播放WAV格式文件，播放完毕后将控制转移回应用中
```

```
Public Const SND_ASYNC=&&H1 ‘播放WAV格式文件，将控制立即转移回应用程序中，而不管对WAV文件的播放是否结束
```

```
Public Const SND_LOOP &&H8 ‘循环播放WAV格式文件
```

```
Private Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" (ByVal lpstrSoundName As String, ByVal uFlags As Long) As Long
```

其次，在窗中激活事件中加入以下代码：

```
Private Sub Form_Activate()
```

```
Dim ReturnValue As Long
```

```
ReturnValue = sndPlaySound("C:TEST.WAV", SND_SYNC)
```

```
End Sub
```

(2) 使用函数mciExecute播放MIDI格式文件

函数mciExecute只有一个参数：lpstrCommand，它是一个命令字符串，用于表明对声音文件播放的命令，例如，要完整播放声音文件“C:Canyon.mid”，只需要用以下代码：

```
Private Declare Function mciExecute Lib "winmm.dll" (ByVal lpstrCommand As String) As Long
```

```
Private Sub Form_Activate()
```

```
Dim ReturnValue As Long
```

```
ReturnSoundValue = mciExecute("play C:Canyon.mid")
```

```
End Sub
```

9. 结论

VB在课件的开发中对窗口的处理还有很多丰富多彩方法，本文所讨论了只是其中的一些。由于VB继承了Basic简单易学的特点，且多媒体功能强大等特点，只要我们不断地学习和实践，就可以借助VB开发出高水平CAI课件。

本文所介绍的程序已在VB6.0中文企业版的环境下运行通过。

参考文献

1. 刘炳文，张坤编著 Visual Basic程序设计 人民邮电出版社 1999
2. 清源计算机工作室著 Visual Basic6.0开发宝典 机械工业出版社 1999
3. 朱从旭 著 VB中几种窗口特效技术的实现方法和示例 现代计算机 1999.10

作者简介：方刚、1961年、男、硕士、副教授

工作单位：广东技术师范学院计算机科学系

地址：广州市石牌华师大教师新村C座2102、邮政编码510631

电话：020-38746780

E-mail: gzfangan@163.com